Dynamic Voltage and Frequency Scaling : The laws of diminishing returns

Authors

Etienne Le Sueur
and
Gernot Heiser

Workshop on Power Aware Computing and Systems, pp. 1–5, Vancouver, Canada, October, 2010

1. *Senior Member of Technical Staff at Vmware.*
   During the research he was at NICTA is Australia's Information and Communications Technology Research Centre of Excellence.
2. *Scientia Professor and*
   John Lions Chair of Operating Systems
   School of Computer science and Engineering
   UNSW , Sydney, Australia

Presented By

Prasanth B L
Aakash Arora

Introduction

- ***What contributions are in this paper (according to the authors)?***

  The Authors analyzed and examined the potential of DVFS across three platforms with recent generation of AMD processors in various aspects viz.,. Scaling of Silicon Transistor technology, Improved memory performance, Improved sleep/ idle mode, Multicore Processors.

- ***What possible consequences can the contributions have?***

  The results shows that on the most recent platform, the effectiveness of DVFS is markedly reduced, and actual savings are only observed when shorter executions (at higher frequency) are padded with the energy consumed when idle.

Previous Works Before Author related to DVFS

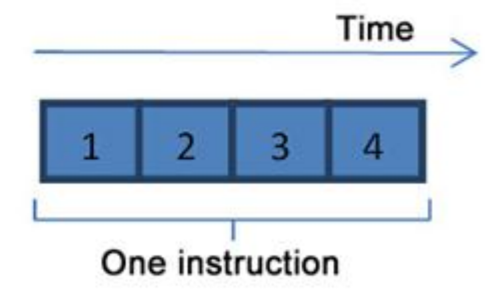
Previous research has attempted to leverage DVFS as a means to improve energy efficiency by lowering the CPU frequency when cycles are being wasted, stalled on memory resources. Energy can only be saved if the power consumption is reduced enough to cover the extra time it takes to run the work load at the lower frequency.

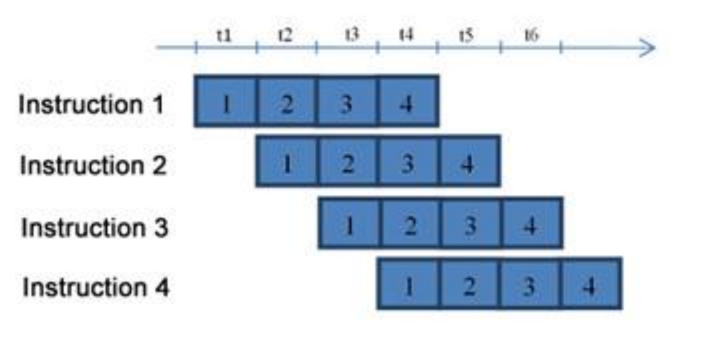
"Hot" gigahertz

Higher the frequency more the power consumption

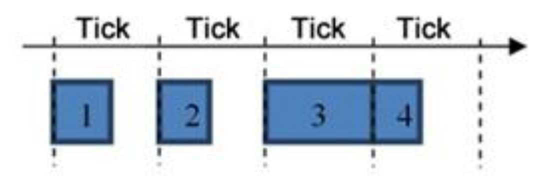
- An example to understand the scaling of Voltage and frequency by simple instruction execution

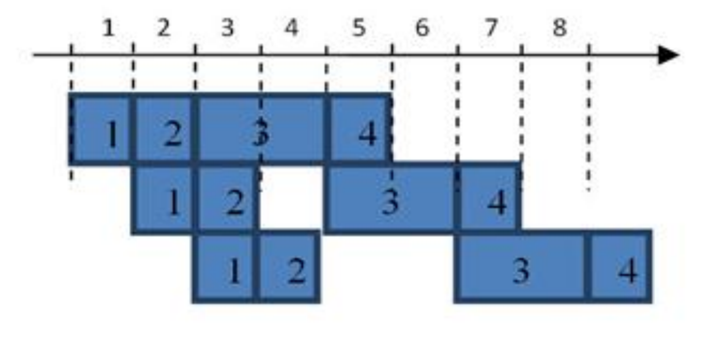
Case 1: Equal execution time for each stage





Case 2: Unequal execution time for each stage





Previous Research works Referenced

- They used simulated execution traces and the level of slack time to choose a new CPU frequency at each OS scheduler invocation [7].

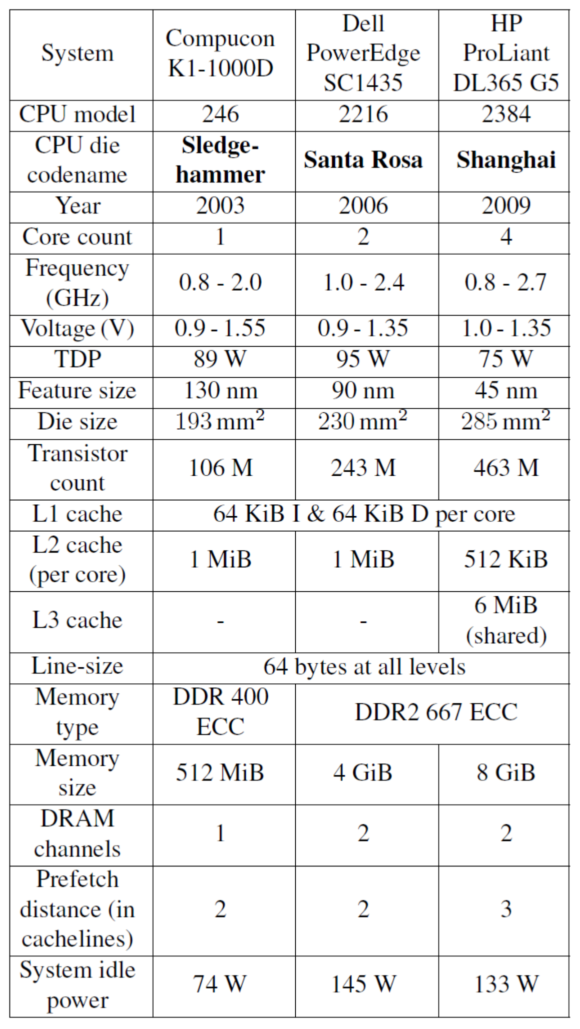
- From the performance monitoring unit (PMU) available in most processors, parameters such as memory request per cycle and instruction per cycle are accounted and predicted work load response and a change in a new CPU frequency at each OS scheduler invocation [8].

- Developing a technique to automatically choose best parameters from the PMU . Koala framework is used to save up to 20 % of energy [5,6] .

- Energy consumed when idle must be accounted for if saving energy is the overall goal [4].

- DVFS still effective on smart phones [1].

Experimental Setup



System	Compucon K1-1000D	Dell PowerEdge SC1435	HP ProLiant DL365 G5
CPU model	246	2216	2384
CPU die codename	**Sledge-hammer**	**Santa Rosa**	**Shanghai**
Year	2003	2006	2009
Core count	1	2	4
Frequency (GHz)	0.8 - 2.0	1.0 - 2.4	0.8 - 2.7
Voltage (V)	0.9 - 1.55	0.9 - 1.35	1.0 - 1.35
TDP	89 W	95 W	75 W
Feature size	130 nm	90 nm	45 nm
Die size	193 mm^2	230 mm^2	285 mm^2
Transistor count	106 M	243 M	463 M
L1 cache	64 KiB I & 64 KiB D per core		
L2 cache (per core)	1 MiB	1 MiB	512 KiB
L3 cache	-	-	6 MiB (shared)
Line-size	64 bytes at all levels		
Memory type	DDR 400 ECC	DDR2 667 ECC	
Memory size	512 MiB	4 GiB	8 GiB
DRAM channels	1	2	2
Prefetch distance (in cachelines)	2	2	3
System idle power	74 W	145 W	133 W

- ***OS :***
  Linux 2.6.33
- ***Power Measurement Device:***
  Extech 380801 AC Power Analyser
- ***Test Benchmark suite:***
  SPEC CPU2000

SPEC CPU2000 Benchmark File Description

- **181.mcf, 429.mcf**

  A benchmark derived from a program used for single-depot vehicle scheduling in public mass transportation. The program is written in C, the benchmark version uses almost exclusively integer arithmetic.

- **These Benchmark files will have the input and output description.**

Parameters Measured

- Runtime , Energy, Energy Delay Product for different frequency of operation.

Power Management

Frequency Scaling:- The processor clock is reduced by some multiple of the maximum, permitting the processor to consume less power at the expense of reduced performance

Clock Throttling:- In contrast with frequency scaling where the frequency of the clock is actually modified, clock throttling keeps the clock running at the original frequency, however, the clock signal is disabled for some number of cycles at regular intervals.

Dynamic Voltage Scaling (DVS):- Reduces the power consumed by lowering its operating voltage.

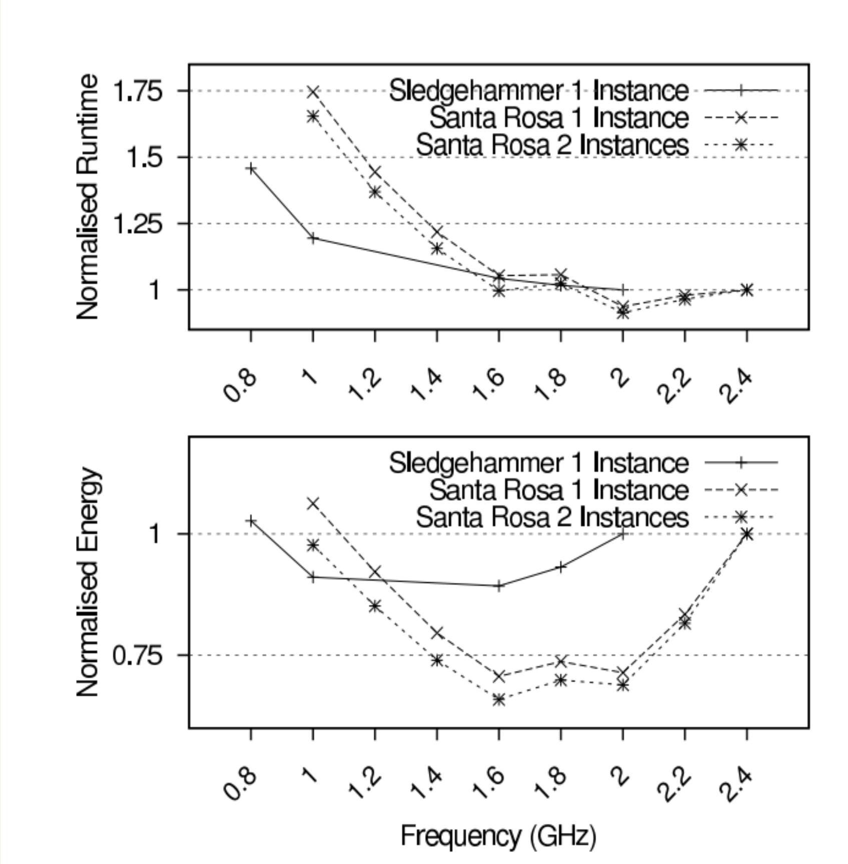
Voltage scaling is advantageous because power consumed by a processor is directly proportional to V^2

Analysis of Results obtained

Normalised runtime (top) and energy consumption (bottom) of 181.mcf at different frequencies on the Sledgehammer and Santa Rosa platforms.

Results obtained from running a single instance of 181.mcf on the Sledgehammer and one or two instances on the Santa Rosa based platform

Energy consumption can be reduced by using DVFS



The Energy/Frequency Convexity Rule: Modeling and Experimental Validation on Mobile Devices by Karel De Vogeleer etal. [2014]

- Provides an Energy/Frequency Convexity Rule, which relates energy consumption and CPU frequency on mobile devices
- Modeling Energy Consumption
- **Model the amount of clock cycles** to complete a benchmark sequence of instructions

$$t = \frac{cc_b}{f^{\beta} - cc_k}$$

t : total time to complete the program

f: system's clock frequency

cc_b: number of clock cycles to complete a benchmark of instructions

β: an architecture-dependent scaling constant

cc_k: number of cycles spent on the OS

- **Power Consumption:**

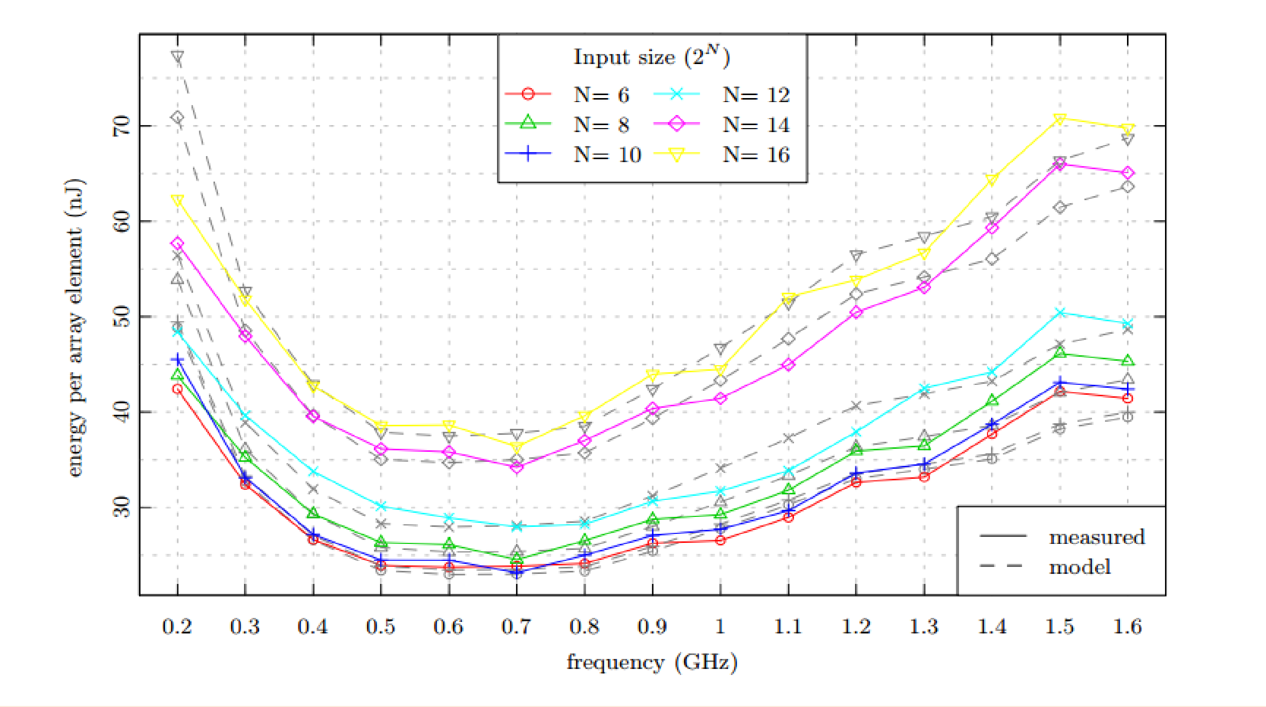
$$\boldsymbol{P = P_{system} + P_{leak} + P_{dynamic}}$$
$$\boldsymbol{P = P_{system} + \gamma V P_{dynamic} + P_{dynamic}}$$
$$\boldsymbol{P = P_{system} + (1 + \gamma V)\eta\alpha C f V^2}$$

Contd…

- **Energy Consumption**

$$E = E_{leak} + E_{dynamic}$$

$$E = (1 + \gamma V)\eta\alpha CfV^2 \frac{cc_b}{f^{\beta} - cc_k}$$



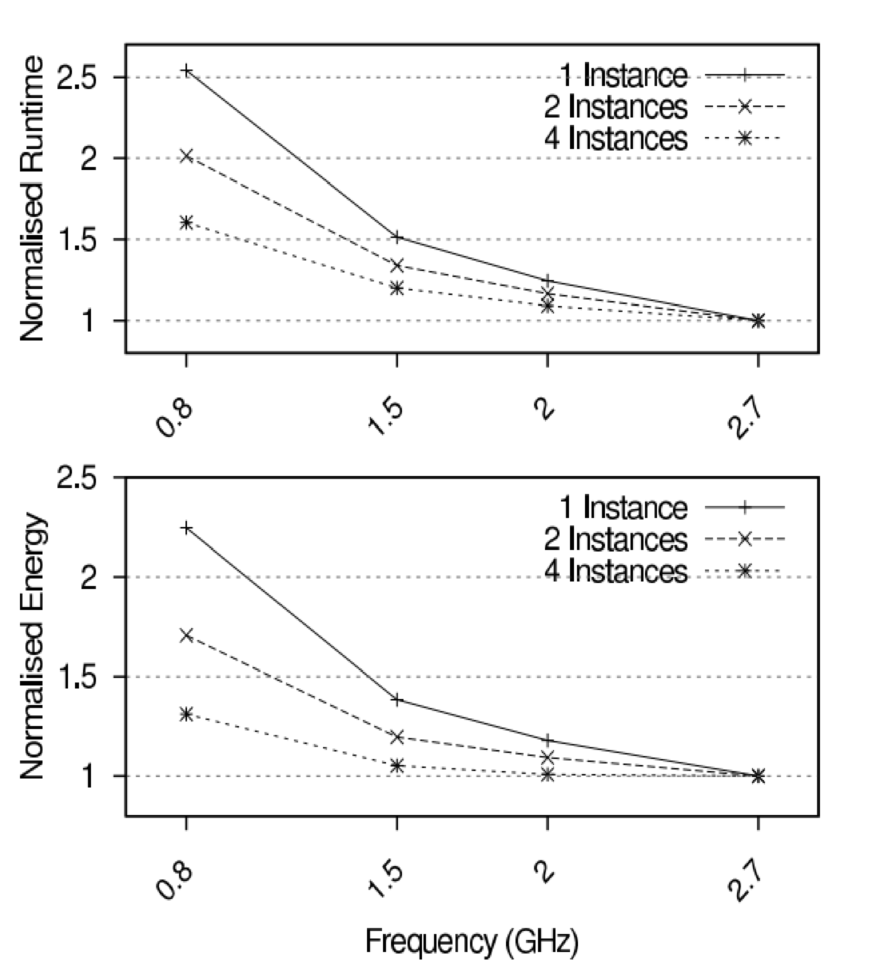
Normalised runtime (top) and energy consumption (bottom) of 181.mcf at different frequencies on the Shanghai platform

Normalised runtime increases with a reduction in frequency

Energy consumption increases with the use of DVFS

For *Santa Rosa* memory frequency vary between [280Hz,333Hz] and on *Shanghai* platform memory frequency is 333 Hz for all CPU frequencies

Lower memory access latency reduces pipeline stalls which in turn reduces the opportunities to save energy using DVFS



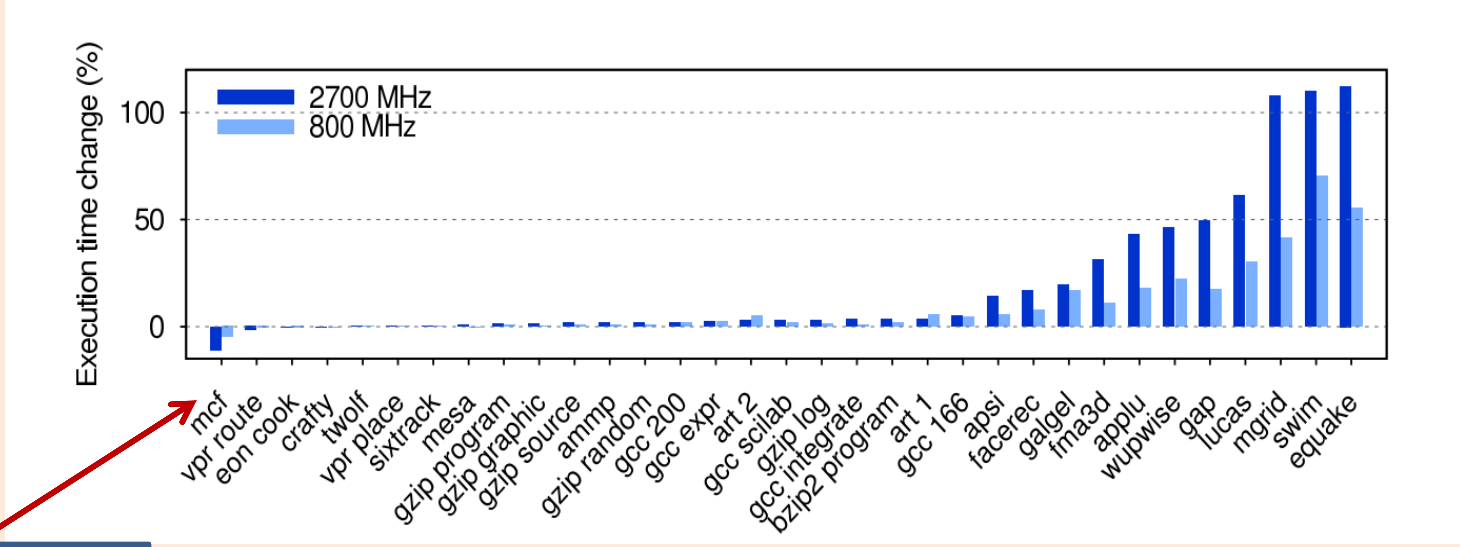
Scaling of Silicon Transistor Technology

- Smaller the Transistor → Lower Threshold voltage→ Increased Sub threshold leakage current i.e. static power is increased.
- Smaller Transistors works at higher frequency with lower supply voltage
- The net effect is a reduction in the dynamic range of power consumption that DVFS can utilize and increase in static power consumption.

Improved Memory Performance

- Memory speed is still improving with respect to a single CPU core. This increases the scope to use pre-fetching to further hide memory access latency by reducing the number of cache misses.
- Lower memory access latency reduces pipeline stalls which in turn reduces the opportunities to save energy using DVFS.

Demonstrates the performance benefit achieved from DRAM prefetching for SPEC CPU2000 workloads on the <u>Shanghai</u> platform



Exception

Some workloads more than double their execution time when prefetching is disabled

This clearly shows how important prefetching is on newer platforms, where a single core cannot issue memory requests fast enough to saturate the memory bus

Some Observations

- No longer see significant increases in the clock speeds of CPUs due to transistor scaling
- Memory speed is still improving with respect to a single CPU core
- This increases the scope to use prefetching
- On the Sledgehammer platform, when the CPU frequency is reduced to 800 MHz, the memory frequency drops from 200 MHz to 160 MHz
- On the Santa Rosa platform, the memory frequency was observed to vary between 280 MHz and 333 MHz depending on the chosen CPU frequency
- So, on the older platforms, memory frequency is dependent on CPU frequency

Improved Sleep / Idle mode

- After a work has been completed, the system stays powered on and goes into an idle state. The depth of sleep determines the power consumption during this period and the latency to wake up.
- Memory also contributes to idle power because DRAM must be refreshed periodically to retain data.
- Processor with large cache will consume more power in the C1 state , because cache must be kept coherent
- CPU is not executing instructions in these C-states

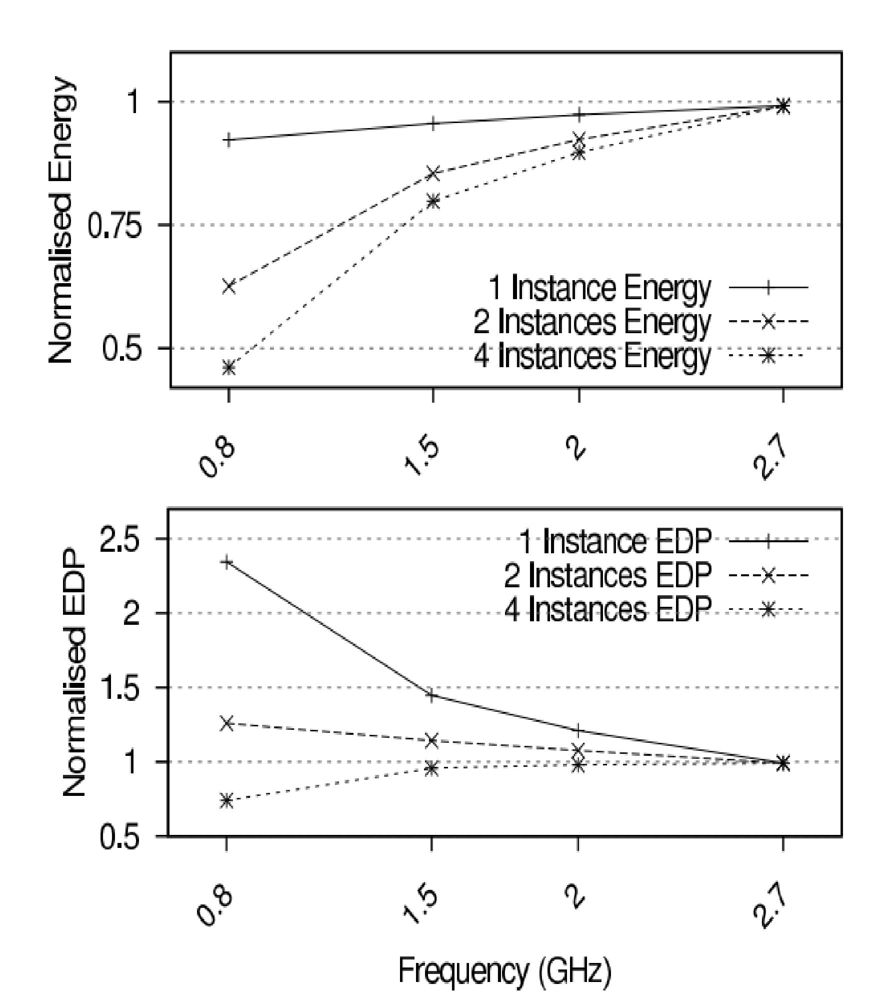
Energy consumption and energy-delay product for 181.mcf on the Shanghai platform when padded with idle energy

DVFS appears to become much more effective when idle power is factored in.

Optimal energy efficiency is achieved by running at the lowest frequency

Find a good balance of energy savings and performance degradation

In such cases the quantity that should be optimised is the energy-delay product (EDP)



Multi core Processors

- Implementation of DVFS in multicore platform is complex
- Chipwide DVFS forces each core on a package to operate at the same frequency and voltage.
- Workloads running on multiple cores must be analysed as a whole in order to determine whether or not to scale frequency
- The AMD opetreon processor they have selected each core can operate at a different frequency but the voltage must be no lower than required by the core operating at the highest frequency.

Conclusion

- Our results show that on the most recent platform, the effectiveness of DVFS is markedly reduced
- Actual savings are only observed when shorter executions (at higher frequencies) are padded” with the energy consumed when idle
- Energy-delay product, which balances energy savings against performance losses, is minimised at the highest frequency in all
- Only considered the effectiveness of DVFS on server-class platforms, while DVFS may still be effective on other platform, such as smart-phones
- “Turbo-Boost” technology can improve performance and energy efficiency by increasing the frequency to complete a workload for a shorter-time and then entering low-power sleep modes

Reference

1. AMD Opteron processor reference. http://products.amd.com/en-us/OpteronCPUResult.aspx.
2. SPEC CPU2000 benchmark. http://www.spec.org/cpu2000, Aug 2000.
3. CARROLL, A., AND HEISER, G. An analysis of power consumption in a smartphone. In 2010 USENIX (Boston, MA, USA, Jun 2010).
4. MIYOSHI, A., LEFURGY, C., HENSBERGEN, E. V., RAJAMONY, R., AND RAJKUMAR, R. Critical power slope: understanding the runtime effects of frequency scaling. In 16th Int. Conf. Supercomp. (New York, NY, USA, Jun 2002), ACM Press, pp. 35–44.
5. SNOWDON, D. C. OS-Level Power Management. PhD thesis,School Comp. Sci. & Engin., University NSW, Sydney 2052,Australia, Mar 2010. Available from publications page at http://www.ertos.nicta.com.au.au/
6. SNOWDON, D. C., LE SUEUR, E., PETTERS, S. M., AND HEISER, G. Koala: A platform for OS-level power management. In 4th EuroSys Conf. (Nuremberg, Germany, Apr 2009).
7. WEISER, M., WELCH, B., DEMERS, A. J., AND SHENKER, S. Scheduling for reduced CPU energy. In 1st OSDI (Monterey, CA,USA, Nov 1994), pp. 13–23.
8. WEISSEL, A., AND BELLOSA, F. Process cruise control—eventdriven clock scaling for dynamic power management. In CASES (Grenoble, France, Oct 8–11 2002).
9. https://software.intel.com/en-us/blogs/2014/02/19/why-has-cpu-frequency-ceased-to-grow